AD-A092 474 SCIENCE APPLICATIONS INC LA JOLLA CA F/G 13/10

PROGRAM MANUAL NSUP.(U)

SEP 80 R B CHAPMAN N00014-78-C-0111

UNCLASSIFIED NL

1 of 1

AD A 092474

END

DATE FILMED

1-81

DTIC

LEVEL IV

AD A092474

DTIC
DEC 3 1980
C

SCIENCE APPLICATIONS, INC.

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED

80 11 24 014

6 Program Manual
NSUP

by

10 Richard B. Chapman

11 Sep[redacted]80 12 35

9 Final rept. Dec 77- Sep 80

This program computes forces and moments and the free surface disturbance generated by an arbitrary three-dimensional ship hull with arbitrary linearized motions. Forward speed is included under the assumption that the flow generated by forward speed can be linearized as well.

15 N00014-78-C-0111

16 SR02301

17 SR0230101

61153N

SCIENCE APPLICATIONS, LA JOLLA, CALIFORNIA
ALBUQUERQUE • ANN ARBOR • ARLINGTON • ATLANTA • BOSTON • CHICAGO • HUNTSVILLE
LOS ANGELES • McLEAN • PALO ALTO • SANTA BARBARA • SUNNYVALE • TUCSON

P.O. Box 2351, 1200 Prospect Street, La Jolla, California 92037

388862

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER None	2. GOVT ACCESSION NO. AD-A092 474	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Program Manual NSUP		5. TYPE OF REPORT & PERIOD COVERED Final - Dec 1977 - Sep 1980
		6. PERFORMING ORG. REPORT NUMBER None
7. AUTHOR(s) Richard B. Chapman		8. CONTRACT OR GRANT NUMBER(s) Contract N00014-78-C-0111
9. PERFORMING ORGANIZATION NAME AND ADDRESS Science Applications, Inc. 1200 Prospect Street La Jolla, CA 92038		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS SR 023 01 01
11. CONTROLLING OFFICE NAME AND ADDRESS David W. Taylor Naval Ship R&D Center, Code 1505 Bethesda, Md 20084		12. REPORT DATE September 1980
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Office of Naval Research 800 N. Quincy Street Arlington, Va 22217		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Sponsored by the Naval Sea Systems Command General Hydromechanics Research Program administered by David W. Taylor Naval Ship R&D Center, Code 1505, Bethesda, Md 20084. Program releated to Science Applications, Inc., Report SAI-462-80-560-LJ

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Ship Motions,
Three-Dimensional Ship Hull
Seakeeping
Hydrodynamics

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This program computes forces and moments and the free surface disturbance generated by an arbitrary three-dimensional ship hull with arbitrary linearized motions. Forward speed is included under the assumption that the flow generated by forward speed can be linearized as well.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

INTRODUCTION

The program NSUP is written in a form which allows data to be entered interactively on a DEC-10 computer system or equivalent. However the program can be easily adapted to batch input and to other computer systems by altering the input and output statements. Also, the velocity history is defined by step functions in each of the six degrees of freedom to aid in computing results in the frequency domain. This restriction can be easily removed and arbitrary velocity histories specified.

Groups of Subroutines

One main program (MAIN) and 19 subroutines make up the program. The nineteen subroutines used in NSUP can be arranged into groups according to their functions as outlined below.

1. Input and Initialization

   Much of the input and initialization is done in the main program (MAIN). In addition the main program calls subroutines EXP, SINP, MTN, AFSTN, and AFSR to aid in this function.

2. Special Function Evaluation

   To save computer time the exponential and trigometric functions are computed from prepared tables. Subroutines EXF and SINQ perform the required interpolation.

3. Computation of the Source Strength Distribution and its Time Derivative

   Subroutine BLANC computes the source strength and its time derivative at the center of each panel from the velocities and accelerations, respectively, at the panel centers relative to the local (free-surface induced) flow.

4. Free-Surface Computations

Subroutines ACPTR, PRFR, and CFSR all involve free-surface related computations.

5. Body Computations

Subroutines EBD, POTB, POTST, SELF, MATJN, GE, GO, SOLID, and PREP all aid in the computing body-related matrices.

The main program and each of the nineteen subroutines are described below.

Program Description

MAIN: Initially this program checks to see if certain large arrays which depend only the body geometry have been computed and stored in their assigned files on previous runs. If so they are read in directly and do not have to be computed. The main program then calls subroutine EDB which reads in the panel description of the hull and computes arrays E and PX if they are not available. The matrix E gives the source distribution vector (or its time derivative) from the vector specifying the normal relative velocities (or accelerations) at the panel centers. Array PX gives the generalized force vector induced by a source distribution vector for finite speed (the $\rho U \frac{\partial \phi_{BD}}{\partial x}$ term in the pressure). If required the matrix P, giving the force vector induced by the time derivatives of the source strengths of the panels, is computed by subroutine POTST. Next subroutine MTN is called to specify the body velocities for impulsive motion in each of the six degrees of freedom. Then AFS and AFSR subroutines are called to define and initialize the free-surface. After initializing the free-surface the main program calls a pair of routines to compute trigometric and exponential tables (SINP and EXP) for later use. The main program

executes the time stepping loop for NTM values of time. The first step is impulsive and covers a time interval of zero duration to obtain impulsive pressures. The forces and moments computed by this initial time step are the impulsive values resulting from starting the body with finite speed. All later time steps are of fixed duration (DT) with forces and moments acting on the body giving a time history over the duration of the calculation.

The first subroutine to be called in each time step is ACPTR which computes the panel pressures and normal accelerations induced by the free surface. Next subroutine ZBLANC subtracts the free-surface component of normal acceleration as computed by ACPTR from the accelerations of the body to obtain the relative acceleration at each panel center. It then applies matrix E to find the time derivative of the body source distribution.

The free-surface induced force vector, PF, is then computed from the pressure distribution by subroutine PRFR (called by ACPTR). Subroutine POTB, called by the main program, computes the force vector PB, induced by the time rate of change of the panel source distribution (through matrix P) and the spatial derivative of the body-induced potential in the x direction for finite forward speed (through matrix PX). The two force vectors, PF and PB, are added to give the total generalized force vector PT for each of the six degrees of freedom. Finally subroutine CFSR is used to advance the free surface by a single time step, completing the loop.

Subroutine EXP: Sets up a table for the exponential function.

Subroutine EXF: Uses the table to compute the exponential quickly.

Subroutine MIN: Reads in impulsive velocities for each degree of freedom, time step size, and number of time steps.

Subroutine SINP: Sets up a table for trigometric functions.

Subroutine SINP: Uses the table to compute sine and cosine functions quickly.

Subroutine ZBLACN: Applies the specified body accelerations for the six degrees of freedom to compute the resulting normal accelerations at the panel centers. The free-surface induced normal accelerations ACNW(J) are subtracted to obtain the net normal acceleration at each panel center, ACN(J), of the body relative to the fluid. These accelerations must be cancelled by the time derivative of the body source density distribution, ST(K). The ACN(J) vector is multiplied by the E matrix to get the necessary net rate of change of the panel source densities, vector ST(K). The total source strength densities are accumulated in STOLD(K).

Subroutine ACPTR: Computes free-surface induced accelerations ACNW(J) and pressures PRFS(T) at panel centers.

Subroutine PRFR: Computes generalized force vector PF for six degrees of freedom from the computed free surface pressure distribution at the center of each panel, PRFS(T).

Subroutine AFSIN: Reads in parameters defining the free-surface representation, then calls AFSR.

Subroutine AFSR: Sets up and initializes the free-surface representation.

Subroutine CFSR: Advances free-surface by one time increment. Moves body relative to the free surface. Adds the changes in free-surface elevation induced by the body sources acting over the time increment to the free-surface representation. Second order effects in time are included.

Subroutine EBD: This subroutine reads in the (x, y, z) coordinates of each of the four corner points into a set of arrays XPT(N), YPT(N), ZPT(N). Panels are identified by a set of four integers giving the array positions of the four corner points of each panel. Panel areas, normals and center point coordinates are then computed. Finally,

the E matrix giving the source time derivative distirbution for a set of prescribed normal accelerations is computed. The inverse of E is computed first by the subroutine GE which gives the acceleration induced at any panel center point, J, by a uniformly distributed time derivative of source strength density of unit magnitude acting over any surface of panel, JL. Subroutine MATIN inverts E to obtain the desired form. Simultaneously, the matrix PX which gives the x component of velocity at the center of panel J induced by a source strength of unit magnitude distributed over a panel, JL, is computed.

Subroutine POTB: This subroutine is used to compute the generalized body-induced force vector, PB, for all six degrees of freedom generated by a known source strength distribution and its time derivative. The matrix P(J,K) is multiplied by a vector, ST(K), representing the time derivative of source densities of the panels to obtain one term of PB. Similarly the term proportional to forward speed is computed from the matrix PX and a vector representing the accumulated source densities, STOLID(K).

Subroutine POTST: Calculates the matrix P(N,J) giving the net force or moment for the Jth degree of freedom induced by a unit time derivative of source strength over panel N. Fundamental to this is the need to compute the potential integrated over each panel area due to a uniform source density over every other panel. For panels which are far apart relative to their dimensions this value is, for unit source density, simply proportional to the product of their areas divided by the distance between centers. The method used here is to divide each panel into a large number of small subpanels and then calculate the result numerically, adding the contributions of each subpanel under the assumption that their separations are large relative to their dimensions.

Subroutine SELF: This subroutine is called by POTST to compute diagonal terms in the P matrix.

Subroutine MATJN: The matrix inversion routine used by subroutine EBD to invert matrix E.

Subroutine GE: A subroutine called by EBD to compute the elements of matrix E prior to inversion. It computes the velocity (acceleration) at field point (XF, YF, ZF) induced by a source density (time rate of change of source strength) of value unity distributed uniformly over panel J.

Subroutine GO: A subroutine called by GE.

Subroutine SOLID: Also called by subroutine GE to compute the solid angle of a panel relative to the field point.

Subroutine PREP: Prepares all panels for the GE subroutine. It is called by EDB prior to using GE.

Sample Input

As an example of the input required to execute NSUP interactively from a remote terminal, a sample input is given below. Asterisks have been placed at the beginning of all lines output by the program to distinguish them from inputs supplied by the user. Comments have also been added. They are distinguished by placing them in parentheses.

```
*TYPE 1 TO READ IN PREVIOUSLY CMPTD BODY ARRAYS
 1
*TYPE 1 TO OUTPUT FREE-SURFACE ACCELERATIONS AND PRESSURES
 1
 (this option gives a printout of the free-surface induced accel-
  erations and pressures computed at the panel centers for every
  time step)
*BODY HAS BEEN DEFINED WITH 60 PANELS
 (start subroutine MTN)
```

```
*HOW MANY TIME STEPS AFTER INITIAL STEP
 150
*WHAT IS THE TIME STEP INTERVAL - ZERO FOR FIRST STEP
 0.10
*BODY VELOCITIES
*VELOCITIES ZERO TIME NEGATIVE - CONSTANT TIME POSITIVE
 (program assumes a velocity step function at time = zero)
*WHAT IS SURGE VELOCITY FOR POSITIVE TIME
 0.
*WHAT IS SWAY VELOCITY FOR POSITIVE TIME
 0.
*WHAT IS HEAVE VELOCITY FOR POSITIVE TIME
 1.0
*WHAT IS  ROLL  VELOCITY FOR POSITIVE TIME
 0.
*WHAT IS PITCH VELOCITY FOR POSITIVE TIME
 0.
*WHAT IS YAW VELOCITY FOR POSITIVE TIME
 0.
 (end subroutine MTN and start subroutine AFSIN)

*INPUT ACCELERATION OF GRAVITY AND FLUID DENSITY
 1.0
 1.0
*WHAT IS FORWARD SPEED?
 0.0
*INPUT MAX X LENGTH SCALE
 2.2
*INPUT MIN X LENGTH SCALE
 0.20
*INPUT MAX Y SCALE
 2.2
*INPUT MIN Y LENGTH SCALE
 0.20
*INPUT MAXIMUM TIME SCALE
 8.0
```

(The time scale is a trade-off between accuracy over the later time steps and computational time. Generally it can be set to a value somewhat less than the duration of the simulation without introducing significant errors.)

(END OF INPUT)

Because of its size and complexity the panels defining the hull are input from a prepared file named PANIN.DAT. The definitions of the variables which are input and the formats by which they are read can be easily seen from the instructions and comments at the beginning of subroutine EBD. Integers and floating point numbers are input with formats 4I5 and 3F10.0, respectively. The first line in the file defines NPT, the number of corner points, and NPAN, the number of panels:

```
READ (23,101) NPT, NPAN .
```

(The file PANIN.DAT has been assigned to unit 23 previously.)

Next the coordinates of the corner points are read in by:

```
READ(23,100)(XPT(N),YPT(N),ZPT(N),N=1,NPT) .
```

Finally the panels are defined by giving the four corner points (in a clockwise sense when viewed from a point outside of the body along the panel normal). These four points are identified by integers specifying the position of the corner point in the corner point array previously read in:

```
READ(23,101)(KK(N,1),KK(N,2),KK(N,3),KK(N,4),N=1,NPAN) .
```

Output

The program starts its output on the terminal with two columns of floating point numbers giving the x wave numbers, kx_n, and their increments, Δkx_n. These columns are repeated for the y wave numbers, ky_m, and their increments, Δky_m. For the case

defined by the sample input, the x and y wave numbers are identical with 19 values each starting at 0.01563 and ending at 5.33523. The total number of modes is 19 x 19 = 361.

During each time step information is typed out. As an example the output during the initial time step of the computation resulting from the sample input is listed below.

IMPULSE FORCE AT T=0+

(only for initial step)

TIME=0.0000000

FREE SURFACE INDUCED FORCES---

0.000000	0.000000	0.000000
0.000000	0.000000	0.000000

(initial values)

ACCELERATIONS

0.00000000	0.00000000	0.00000000	0.00000000	0.00000000

(repeated over 12 lines total)

PRESSURES

0.00000000	0.00000000	0.00000000	0.00000000	0.00000000

(Again repeated over 12 lines. Initially, the free-surface disturbance is zero and does not induce any pressure or accelerations at the sixty panel centers)

BODY INDUCED FORCES=

-0.000030	0.006556	-1.015025
0.000039	-0.000109	-0.000124

TOTAL FORCE COMPONENTS

-0.000030	0.006556	-1.015025
0.000039	-0.000109	-0.000124

(END OUTPUT ON TERMINAL)

In addition to the output on the terminal, a file named TOTAL.DAT is created containing the times and the total force components computed at each time step. Also if the arrays E, P, and PX must be computed, they are stored on permanent files for future runs.

Program Listing

A listing of the programs is given over the next few pages.

```
C MAIN PROGRAM
      COMMON/BD/XPAN(120),YPAN(120),ZPAN(120),AREA(120),ST(120),
     .ACN(120),ACNW(120),AN(120,3),E(120),P(120,6),PRFS(120),STOLD(120)
     .,PX(120,6)
      COMMON/FS/AKZ(30,30),SS(30,30),CC(30,30),
     .DKX(30),DKY(30),AKX(30),AKY(30)
      COMMON/BD2/XPT(150),YPT(150),ZPT(150),WRF(150),WRFR(150)
     .,KK(150,4)
      COMMON/A/NPAN,NPT,GEE,RHO,NKX,NKY,EYE,DT,TIM,UFWD
      DIMENSION PF(6),PB(6),PT(6)
      COMPLEX EYE
       EYE=(0.0,1.0)
      TYPE 4442
 4442 FORMAT(' TYPE 1 TO READ IN PREVIOUSLY CMPTD BODY ARRAYS')
      ACCEPT 400, NSKB
  400 FORMAT(I5)
C  COMPUTE BODY MATRICES E AND PX
       CALL EBD(NSKB)
      TYPE 1111
 1111 FORMAT(' TYPE 1 TO OUTPT FREE-SURFCE ACCELERATIONS AND PRESSURES')
      ACCEPT 400,NTVP
      IF(NSKB.NE.1)  CALL POTST
      IF(NSKB.NE.1) GO TO 7334
C     READ PRESSURE MATRICES INTO CORE
      OPEN(UNIT=1,FILE='PTSV',ACCESS='SEQIN',DEVICE='DSK:')
      OPEN(UNIT=2,FILE='PXSV',ACCESS='SEQIN',DEVICE='DSK:')
      DO 544 K=1,6
      READ(1)(P(J,K),J=1,NPAN)
  544 READ(2)(PX(J,K),J=1,NPAN)
      CLOSE(UNIT=1)
      CLOSE(UNIT=2)
 7334 NTM=0
       TYPE 16,NPAN
   16 FORMAT(' BODY HAS BEEN DEFINED WITH',I5,'PANELS')
C OPEN FILE TO STORE TOTAL FORCE COMPONENTS
      OPEN(UNIT=19,FILE='TOTAL',ACCESS='SEQOUT',DEVICE='DSK:')
C IMPULSIVE BODY VELOCITIES (CONSTANT AFTER IMPULSIVE FIRST STEP)
      CALL MTN(NTM,VXMT,VYMT,VZMT,VRLMT,VPMT,VYWMT)
C   INITIALIZE FREE SURFACE
      CALL AFSIN(T,BGX,SMX,BGY,SMY)
C  INITIALIZE SPECIAL FUNCTION GENERATORS
      CALL SINP
      CALL EXP
C SET TIME TO ZERO AND BEGIN LOOP IN TIME(JTM)
      TIM=0.00
       TYPE 2626
 2626 FORMAT(' IMPULSE FORCE AT T=0+')
      DO 157 JTMM=1,NTM
      JTM=JTMM
      IF (JTM.EQ.2) TYPE 2627
 2627 FORMAT(' F(TILDA) IN TIME DOMAIN-- RESPONSE TO STEP FUNCTION ')
      TYPE 17,TIM
      WRITE(19,2121)TIM
   17 FORMAT(//' TIME=',F15.6)
C COMPUTE PANEL PRESSURES AND NORMAL ACCELERATIONS FROM F.S.
      CALL ACPTR(PF,NTVP)
C     FIND SOURCE STRENGTHS OF PANELS
      IF(JTM.NE.1) GO TO 10
C FIRST STEP IS IMPULSIVE VELOCITIES ARE FINAL VALUES
      CALL ZBLACH(VXMT,VYMT,VZMT,VRLMT,VPMT,VYWMT,JTM)
C INITIALIZE SOURCE STRENGTHS
```

```
      DO 4176 J=1,NPAN
 4176 STOLD(J)=ST(J)
      GO TO 20
C ZERO ACCELERATIONS AFTER FIRST IMPULSIVE TIME STEP
C     FIRST STEP IS IMPULSIVE- VELOCITIES REACH FULL VALUE INSTANTLY
   10 CALL ZBLACN(0.,0.,0.,0.,0.,0.,JTM)
   20 CONTINUE
C COMPUTE FORCES CONTRIBUTED BY BODY SOURCE DISTRIBUTION
      CALL POTB(PB)
C   ADD FREE-SURFACE AND BODY INDUCED PRESSURES
      DO 768 NGGN=1,6
      PT(NGGN)=PF(NGGN)+PB(NGGN)
  768 CONTINUE
      TYPE 4433
 4433 FORMAT(' TOTAL FORCE COMPONENTS')
      TYPE 2121,PT(1),PT(2),PT(3),PT(4),PT(5),PT(6)
      WRITE(19,2121)PT(1),PT(2),PT(3),PT(4),PT(5),PT(6)
 2121 FORMAT(3X,3F15.6)
  306 CONTINUE
C   ADVANCE FREE SURFACE
      IF(JTM.EQ.1) GO TO 157
      CALL CFSR(JTM)
      IF(JTM.GT.1) TIM=TIM+DT
  157 CONTINUE
      CLOSE(UNIT=19)
      STOP
      END
```

```
      SUBROUTINE EXP
C PREPARE TABLE OF EXPONENTIAL FUNCTION
      COMMON/EX/EXST(100)
      DEI=EXP(-0.10)
      DET=EXP(-1.0)
      DA=1.0
      J=0
      DO 974 M=1,10
      DB=DA
      DA=DA*DET
      DO 974 N=1,10
      J=J+1
      DB=DB*DEI
      EXST(J)=DB
  974 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE NTM(NTM,V1,V2,V3,V4,V5,V6)
      COMMON/A/NPAN,NPT,GEE,RHO,NKX,NKY,EYE,DT,TIM,UFWD
      COMPLEX EYE
C     READS IN BODY VELOCITY STEP FNCTN (RIGID BODY MOTION)
C STARTING AT TIME=ZERO
         TYPE 10
      ACCEPT 100,NTM
        NTM=NTM+1
      TYPE 300
      ACCEPT 200,DT
      TYPE 9
       TYPE 6
      TYPE 11
      ACCEPT 200,V1
      TYPE 12
       ACCEPT 200,V2
      TYPE 13
      ACCEPT 200,V3
      TYPE 14
      ACCEPT 200,V4
      TYPE 15
      ACCEPT 200,V5
      TYPE 16
      ACCEPT 200,V6
   10 FORMAT(' HOW MANY TIME STEPS AFTER INITIAL STEP')
  100 FORMAT(I5)
  200 FORMAT(F10.0)
  300 FORMAT(' WHAT IS TIME STEP INTERVAL -ZERO FOR FIRST STEP')
   11 FORMAT(' WHAT IS SURGE VELOCITY FOR POSITIVE TIME')
   12 FORMAT(' WHAT IS SWAY VELOCITY FOR POSITIVE TIME')
   13 FORMAT(' WHAT IS HEAVE VELOCITY FOR POSITIVE TIME')
   14 FORMAT(' WHAT IS ROLL VELOCITY FOR POSITIVE TIME')
   15 FORMAT(' WHAT IS PITCH VELOCITY FOR POSITIVE TIME')
   16 FORMAT(' WHAT IS YAW VELOCITY FOR POSITIVE TIME')
    9 FORMAT(' BODY VELOCITIES ')
    6 FORMAT(' VELOCITIES ZERO TIME NEGATIVE-CONST TIME POSITIVE')
      RETURN
      END
```

```
      SUBROUTINE SINP
      COMMON/SNC/SNST(402),CST(402)
      DF=6.283185307/400.0
      F=0.00
      DO 15 N=1,402
      SNST(N)=SIN(F)
      CST(N)=COS(F)
      F=F+DF
   15 CONTINUE
      RETURN
      END
```

```
SUBROUTINE SINQ(SN,CS,ARG)
COMMON/SNC/SNST(402),CST(402)
 AE=ARG*0.159154943092
 NAE=AE
 AE=AE-NAE
 IF(AE.LT.0.0)AE=1.00+AE
 ND=AE*400.0+0.50
 NC=ND+1
DAH=(AE-ND*0.0025)*3.141592654
DAF=DAH+DAH
SN1=SNST(NC)
CS1=CST(NC)
SN=SN1+DAF*(CS1-SN1*DAH)
 CS=CS1-DAF*(SN1+CS1*DAH)
RETURN
END
```

```
      SUBROUTINE ZBLACN(ACBX,ACBY,ACBZ,ACBRL,ACBP,ACBYW,JTMD
C     COMPUTES SOURCE STRENGTHS SATISFYING BODY BOUNDARY CNDTN
      COMMON/BD/XPAN(120),YPAN(120),ZPAN(120),AREA(120),ST(120),
     .ACN(120),ACNW(120),AN(120,3),E(120),P(120,6),PRFS(120),STOLD(120)
     .,PX(120,6)

      COMMON/A/NPAN,NPT,GEE,RHO,NKX,NKY,EYE,DT,TIM,UFWD

      COMPLEX EYE
      OPEN(UNIT=21,FILE='E',DEVICE='DSK:',ACCESS='SEQIN')
D     TYPE 6
D   6 FORMAT(' START ITERATION FOR TIME DERIV OF BDY SOURCE STRENGTHS')
      DO 1800 J=1,NPAN
      X=XPAN(J)
      Y=YPAN(J)
      Z=ZPAN(J)
      ACX=ACBX+ACBP*Z-ACBYW*Y
      ACY=ACBY+ACBYW*X-ACBRL*Z
      ACZ=ACBZ+ACBRL*Y-ACBP*X
C DOT PRODUCT OF BDY ACEL+ BDY NRML FREE-SRFC INDUCED NRML ACEL
      ACN(J)=-ACNW(J)+ACX*AN(J,1)+ACY*AN(J,2)+ACZ*AN(J,3)
C  INTEGRATE TIME DERIVATIVE FOR TOTAL SOURCE STRENGTH
C  AT START OF TIME STEP
      IF(JTM.GT.2) STOLD(J)=STOLD(J)+DT*ST(J)
      ST(J)=0.00
 1800 CONTINUE
D     TYPE 60
D  60 FORMAT('NORMAL ACCELERATIONS')
D       TYPE 15,(ACN(N),N=1,NPAN)
D  15 FORMAT(1X,5F12.6)
  457 REWIND 21
C READ IN INVERSE BODY MATRIX
      DO 1500 J=1,NPAN
      READ(21)(E(JJ),JJ=1,NPAN)
      AC=ACN(J)
      DO 1400 K=1,NPAN
 1400 ST(K)=ST(K)+E(K)*AC
 1500 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ACPTR(PF,NTPAC)
C COMPUTES WAVE ACCELERATIONS AND  PRESSURES  AT
C      PANEL CENTERS
      COMMON/BD/XPAN(120),YPAN(120),ZPAN(120),AREA(120),ST(120),
     .ACN(120),ACNW(120),AN(120,3),E(120),P(120,6),PRFS(120),STOLD(120)
     .,PX(120,6)

      COMMON/FS/AKZ(30,30),SS(30,30),CC(30,30),
     .DKX(30),DKY(30),AKX(30),AKY(30)
      COMMON/FS1/A(30,30),B(30,30),AS(30,30),BS(30,30)

      COMMON/A/NPAN,NPT,GEE,RHO,NKX,NKY,EYE,DT,TIM,UFWD

      COMPLEX EYE,SC,BX,CX,BY,CY,BYCON,SCON,SX,SKON,B1,B2,C1,C2
      COMPLEX A,B,AS,BS

      DIMENSION BX(30),CX(30),PF(6),DX(30)
      DO 1500 J=1,NPAN
      AX=AN(J,1)
      AY=AN(J,2)
      AZ=AN(J,3)
       X=XPAN(J)
      Y=YPAN(J)
      Z=ZPAN(J)
      ACT=0.00
      PRT=0.00
      DVDZ=0.00
      DO 16 N=1,NKX
      CCXX=AKX(N)*X
      CALL SINC(S,C,CCXX)
      BX(N)=CMPLX(C,S)*DKX(N)
      DX(N)=-AKX(N)*AKX(N)*AX
   16 CX(N)=AKX(N)*AX*EYE
      DO 163 M=1,NKY
      CCYY=AKY(M)*Y
      CALL SINC(S,C,CCYY)
      BY=CMPLX(C,S)*DKY(M)
      BYCON=CONJG(BY)
      CY=AKY(M)*AY*EYE
      DO 163 N=1,NKX
      ARGZ=AKZ(N,M)*Z
      CALL EXF(DEP,ARGZ)
      B1=DEP*BY*BX(N)
      B2=DEP*BYCON*BX(N)
      CZ=AKZ(N,M)*AZ
      SC=B1*A(N,M)
      SCON=B2*AS(N,M)
      SX=B1*B(N,M)
      SKON=B2*BS(N,M)
      C1=CX(N)+CY-CZ
      C2=CX(N)-CY-CZ
      ACT=ACT-C1*SC-C2*SCON
      DVZ=DVZ+EYE*AKX(N)*(C1*SX+C2*SKON)/SQRT(AKZ(N,M))
      PRT=PRT+SC+SCON
  163 CONTINUE
C NORMAL ACCELERATION INDUCED BY FREE SURFACE
      DVDZ=DVDZ*UFWD*SQRT(GEE)*DVDZ
      ACNW(J)=ACT*GEE+DVDZ
C  PRESSURE INDUCED BY FREE SURFACE
      PRFS(J)=PRT*GEE*RHO
 1500 CONTINUE
      CALL PRFR(PF)
   80 FORMAT(' ACCELERATIONS')
   60 FORMAT(' PRESSURES')
 1070 FORMAT(1X,5F16.8)
      IF(NTPAC.NE.1) RETURN



      TYPE 80
      TYPE 1070,(ACNW(J),J=1,NPAN)
      TYPE 60
      TYPE 1070,(PRFS(J),J=1,NPAN)
      RETURN
      END
```

```
      SUBROUTINE PRFR(PF)
C COMPUTE FORCES AND MOMENTS DUE TO FREE SURFACE INDUCED PRESSURES
      COMMON/BD/XPAN(120),YPAN(120),ZPAN(120),AREA(120),ST(120),
     .ACN(120),ACNW(120),AN(120,3),E(120),P(120,6),PRFS(120),STOLD(120)
     .,PX(120,6)

      COMMON/A/NPAN,NPT,GEE,RHO,NKX,NKY,EYE,DT,TIM,UFWD

      DIMENSION PF(6)
      COMPLEX EYE
      X1=0.0
      X2=0.00
      X3=0.0
      X4=0.00
      X5=0.00
      X6=0.00
      DO 725 J=1,NPAN
      PRS=PRFS(J)*AREA(J)
      FRX=-AN(J,1)*PRS
      FRY=-AN(J,2)*PRS
      FRZ=-AN(J,3)*PRS
      XF=XPAN(J)
      YF=YPAN(J)
      ZF=ZPAN(J)
      X1=X1+FRX
      X2=X2+FRY
      X3=X3+FRZ
      X4=X4+YF*FRZ-ZF*FRY
      X5=X5+ZF*FRX-XF*FRZ
      X6=X6+XF*FRY-YF*FRX
  725 CONTINUE
       TYPE 80
   80 FORMAT(' FREE SURFACE INDUCED FORCES---')
      TYPE 40,X1,X2,X3,X4,X5,X6
      PF(1)=X1
      PF(2)=X2
      PF(3)=X3
      PF(4)=X4
      PF(5)=X5
      PF(6)=X6
   40 FORMAT(5X,3F15.7)
      RETURN
      END
```

```
      SUBROUTINE AFSIN(T,BGX,SMX,BGY,SMY)
C INPUT FREE SURFACE PARAMETERS

      COMMON/FS/AKZ(30,30),SS(30,30),CC(30,30),
     .DKX(30),DKY(30),AKX(30),AKY(30)

      COMMON/A/NPAN,NPT,GEE,RHO,NKX,NKY,EYE,DT,TIM,UFWD

      COMPLEX EYE
      TYPE 171
      ACCEPT 100,GEE,RHO
      TYPE 200
      ACCEPT 100,UFWD
      TYPE 180
      ACCEPT 100,BGX
      TYPE 181
      ACCEPT 100,SMX
      TYPE 182
      ACCEPT 100,BGY
      TYPE 183
      ACCEPT 100,SMY
      TYPE 20
      ACCEPT 100,T
      CALL AFSR(T,BGX,SMX,BGY,SMY)
      TYPE 201,NKX,NKY
  201 FORMAT(2X,2I5)
      TYPE 21
      DO 152 N=1,NKX
  152 TYPE 102,AKX(N),DKX(N)
      TYPE 22
      DO 153 N=1,NKY
  153 TYPE 102,AKY(N),DKY(N)
  171 FORMAT(' INPUT ACCELERATION OF GRAVITY AND FLUID DENSITY')
  100 FORMAT(F10.0)
  180 FORMAT(' INPUT MAX X LENGTH SCALE')
  181 FORMAT(' INPUT MIN X LENGTH SCALE')
  182 FORMAT('  INPUT MAX Y SCALE')
  183 FORMAT('  INPUT MIN Y LENGTH SCALE')
   20 FORMAT(' INPUT MAXIMUM TIME SCALE')
   21 FORMAT(' X WAVE NUMBERS')
   22 FORMAT(' Y WAVE NUMBERS')
  102 FORMAT(1X,3F12.5)
  200 FORMAT(' WHAT IS FORWARD SPEED?')
      RETURN
      END
```

```
      SUBROUTINE AFSR(T,BGX,SMX,BGY,SMY)
C INTIIALIZES FREE SURFACE
      COMMON/BD/XPAN(120),YPAN(120),ZPAN(120),AREA(120),ST(120),
     .ACN(120),ACNX(120),AN(120,3),E(120),P(120,6),PRFS(120),STOLD(120)
     .,PX(120,6)

      COMMON/FS/AKZ(30,30),SS(30,30),CC(30,30),
     .DKX(30),DKY(30),AKX(30),AKY(30)
      COMMON/FS1/A(30,30),B(30,30),AS(30,30),BS(30,30)


      COMMON/A/NPAN,NPT,GEE,RHO,NKX,NKY,EYE,DT,TIM,UFWD

      COMPLEX A,B,AS,BS,EYE
      DKTT=1.0/(GEE*T*T)
       NBX=0.5/(BGX*DKTT)
       NBY=0.5/(BGY*DKTT)
      AMX=1.0/SMX
      AMY=1.0/SMY
      N=0
      M=0
      AOLD=0.00
    1 IF(N.GE.NBX) GO TO 2
      N=N+1
      AKX(N)=N*N*DKTT
      AOLD=AKX(N)
      IF(AOLD.GE.AMX) NBX=N
      GO TO 1
    2 N=N+1
      AKX(N)=AOLD+1.0/BGX
      AOLD=AKX(N)
      IF(AOLD.LE.AMX) GO TO 2
      NKX=N
      AKX(NKX+1)=AOLD+1.0/BGX
      AOLD=0.00
   11 IF(M.GE.NBY) GO TO 22
      M=M+1
      AKY(M)=M*M*DKTT
      AOLD=AKY(M)
      IF(AOLD.GE.AMY) NBY=M
      GO TO 11
   22 M=M+1
      AKY(M)=AOLD+1.0/BGY
      AOLD=AKY(M)
      IF(AOLD.LE.AMY) GO TO 22
      NKY=M
      AKY(M+1)=AKY(M)+1.0/BGY
      AFF=0.000
      DO 17 N=1,NKX
      AF=(AKX(N+1)+AKX(N))*0.50
      DKX(N)=AF-AFF
   17 AFF=AF
      BFF=0.00
      DO 18 N=1,NKY
      BF=(AKY(N+1)+AKY(N))*0.50
      DKY(N)=BF-BFF
   18 BFF=BF
      DO 100 N=1,NKX
       DO 100 M=1,NKY
      A(N,M)=(0.0,0.0)
      B(N,M)=(0.0,0.0)
      AS(N,M)=(0.0,0.0)
      BS(N,M)=(0.0,0.0)
      AKZ(N,M)=SQRT(AKX(N)**2+AKY(M)**2)
      SIG=SQRT(GEE*AKZ(N,M))
      SS(N,M)=SIN(SIG*DT)


      CC(N,M)=COS(SIG*DT)
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CFSR(JTM)
C ADVANCES FREE SURFACE WAVE SPECTRA IN TIME
      COMMON/BD/XPAN(120),YPAN(120),ZPAN(120),AREA(120),ST(120),
     .ACN(120),ACNW(120),AN(120,3),E(120),P(120,6),PRFS(120),STOLD(120)
     .,PX(120,6)

      COMMON/FS/AKZ(30,30),SS(30,30),CC(30,30),
     .DKX(30),DKY(30),AKX(30),AKY(30)
      COMMON/FS1/A(30,30),B(30,30),AS(30,30),BS(30,30)
      COMMON/A/NPAN,NPT,GEE,RHO,NKX,NKY,EYE,DT,TIM,UFWD

      DIMENSION CX(30)
      COMPLEX EYE,AT,ATS,CX,CY,CXY1,CXYS,A,B,AS,BS,DFWD
C FIRST TIME  STEP IS PURE IMPULSE-  TIME=0.0 AFTER FIRST TIME STEP
      IF(JTM.EQ.1) GO TO 6
      DO 100 N=1,NKX
      CXX=AKX(N)*UFWD*DT
      CALL SINQ(S,C,CXX)
      DFWD=CMPLX(C,-S)
      DO 100 M=1,NKY
      CT=CC(N,M)
      STT=SS(N,M)
      AT=A(N,M)*CT+B(N,M)*STT
      ATS=AS(N,M)*CT+BS(N,M)*STT
      B(N,M)=B(N,M)*CT-A(N,M)*STT
      BS(N,M)=BS(N,M)*CT-AS(N,M)*STT
      A(N,M)=AT
      AS(N,M)=ATS
C    MOVE FREE SURFACE RELATIVE TO BODY WITH FWD SPEED
      A(N,M)=A(N,M)*DFWD
      B(N,M)=B(N,M)*DFWD
      AS(N,M)=AS(N,M)*DFWD
      BS(N,M)=BS(N,M)*DFWD
  100 CONTINUE
    6 CONTINUE
C  NOW ADD EFFECTS OF SOURCE PANELS ACTING OVER ONE TIME STEP
      DO 1500 J=1,NPAN
       STAR=(ST(J)*0.50*DT+STOLD(J))*DT
C ST IS TIME RATE OF CHNGE OF SOURCE STRENGTH
C STOLD IS SOURCE STRENGTH AT START OF TIME STEP
C STAR IS AVERAGE VALUE OF STRENGTH OVER TIME STEP
      STAR2=STOLD(J)*DT
      IF(JTM.EQ.1)STAR=ST(J)
      IF(JTM.EQ.1)STAR2=0.00
      STAR=STAR*AREA(J)*0.6366197724
      STAR2=STAR2*AREA(J)*0.31831
      X=XPAN(J)
      IF(JTM.GT.1) X=X-UFWD*DT*.50
      Y=YPAN(J)
      Z=ZPAN(J)
      DO 93 N=1,NKX
      CXX=AKX(N)*X
      CALL SINQ(S,C,CXX)
   93 CX(N)=CMPLX(C,-S)
      DO 94 M=1,NKY
      CYY=AKY(M)*Y
      CALL SINQ(S,C,CYY)
      CY=CMPLX(C,-S)
      DO 94 N=1,NKX
      ARGZ=AKZ(N,M)*Z
      CALL EXF(DEP,ARGZ)
      CXY1=CX(N)*CY*DEP
       CXYS=CX(N)*DEP*CONJG(CY)
      A(N,M)=A(N,M)+STAR*CXY1
      AS(N,M)=AS(N,M)+STAR*CXYS
C B(N,M) AND BS(N,M) INCREMENTED NEGLECTING CHANGES IN SOURCE STRENGTH


C  OVER TIME INTERVAL ARE SECOND ORDER IN DT
      B(N,M)=B(N,M)-STAR2*CXY1*SS(N,M)
      BS(N,M)=BS(N,M)-STAR2*CXYS*SS(N,M)
   94 CONTINUE
 1500 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE EBD(NSKB)
C INTITIALIZE PANELS AND COMPUTE BODY MATRIX
C
      COMMON/BD/XPAN(120),YPAN(120),ZPAN(120),AREA(120),ST(120),
     .ACN(120),ACNW(120),AN(120,3),E(120),P(120,6),PRFS(120),STOLD(120)
     .,PX(120,6)



      COMMON/BD2/XPT(150),YPT(150),ZPT(150),WRF(150),WRFR(150)
     .,KK(150,4)

      COMMON/A/NPAN,NPT,GEE,RHO,NKX,NKY,EYE,DT,TIM,UFWD

      DIMENSION EP(120),EPP(120)

      COMPLEX EYE
C  READ IN BODY PANEL PARAMETERS
       OPEN(UNIT=23,FILE='PANIN',DEVICE='DSK:',ACCESS='SEQIN')
  101 FORMAT(4I5)
  100 FORMAT(3F10.0)
C  NUMBER OF POINTS AND PANELS
      READ(23,101) NPT,NPAN
C   COORDINATES OF POINTS
      READ(23,100) (XPT(N),YPT(N),ZPT(N),N=1,NPT)
C DEFINE CORNER POINTS OF EACH PANEL
      READ(23,101)(KK(N,1),KK(N,2),KK(N,3),KK(N,4),N=1,NPAN)
      CLOSE(UNIT=23)
C COMPUTE PANEL AREAS
      DO 150 J=1,NPAN
      K1=KK(J,1)
      K2=KK(J,2)
      K3=KK(J,3)
      K4=KK(J,4)
      IF(K4.EQ.0) GO TO 8
      XPAN(J)=(XPT(K1)+XPT(K2)+XPT(K3)+XPT(K4))*0.25
      YPAN(J)=(YPT(K1)+YPT(K2)+YPT(K3)+YPT(K4))*0.25
      ZPAN(J)=(ZPT(K1)+ZPT(K2)+ZPT(K3)+ZPT(K4))*0.25
      GO TO 9
C TRIANGULAR PANELS
    8 XPAN(J)=(XPT(K1)+XPT(K2)+XPT(K3))/3.00
      YPAN(J)=(YPT(K1)+YPT(K2)+YPT(K3))/3.00
      ZPAN(J)=(ZPT(K1)+ZPT(K2)+ZPT(K3))/3.00
      K4=K3
    9 XA=XPT(K3)-XPT(K1)
      XB=XPT(K4)-XPT(K2)
      YA=YPT(K3)-YPT(K1)
      YB=YPT(K4)-YPT(K2)
      ZA=ZPT(K3)-ZPT(K1)
      ZB=ZPT(K4)-ZPT(K2)
C  COMPUTE PANEL AREAS
      AZ=XA*YB-YA*XB
      AX=YA*ZB-ZA*YB
      AY=ZA*XB-XA*ZB
      ARE=SQRT(AX*AX+AY*AY+AZ*AZ)
      AREA(J)=ARE*0.50
      AN(J,1)=-AX/ARE
      AN(J,2)=-AY/ARE
      AN(J,3)=-AZ/ARE
  150 CONTINUE
  806 FORMAT(' J',9X,'NX',9X,'NY',9X,'NZ',9X,'XP',9X,'YP',9X,'ZP',
     .9X,'AREA')
  807 FORMAT(I5,7F11.4)
      IF(NSKB.EQ.1) RETURN
C RETURN IF E AND PX ARRAYS HAVE BEEN COMPUTED IN EARLIER RUNS WITH
C      SAME BODY
```

```
      OPEN(UNIT=21,FILE='E',ACCESS='SEQOUT',DEVICE='DSK:')
      OPEN(UNIT=24,FILE='PXSV',ACCESS='SEQOUT',DEVICE='DSK:')
      DO 1308 J=1,NPAN
       JJJ=J
      CALL PREP(JJJ)
      ST(J)=0.00
      DO 1308 K=1,6
      PX(J,K)=0.00
 1308 CONTINUE
      DO 308 J=1,NPAN
      JJJ=J
      AX=AN(J,1)
      AY=AN(J,2)
      AZ=AN(J,3)
      XF=XPAN(J)
      YF=YPAN(J)
      ZF=ZPAN(J)
      DO 157 L=1,NPT
      WRF(L)=SQRT((XPT(L)-XF)**2+(YPT(L)-YF)**2+(ZPT(L)-ZF)**2)
  157 WRFR(L)=SQRT((XPT(L)-XF)**2+(YPT(L)-YF)**2+(ZPT(L)+ZF)**2)
      DO 309   JL=1,NPAN
      JLJ=JL
       CALL GE(XF,YF,ZF,JLJ,VX,VY,VZ,VXR,VYR,VZR,JJJ)
      VX=VX+VXR
      VY=VY+VYR
      VZ=VZ+VZR
C   COMPUTE NORMAL VELOCITY AT PANEL J DUE TO PANEL JL
      E(JL)=AX*VX+AY*VY+AZ*VZ
C INCREMENTT PX MATRIX
      FR1=-AREA(J)*VX*AN(J,1)
      FR2=-AREA(J)*VX*AN(J,2)
      FR3=-AREA(J)*VX*AN(J,3)
      PX(JL,1)=PX(JL,1)+FR1
      PX(JL,2)=PX(JL,2)+FR2
      PX(JL,3)=PX(JL,3)+FR3
      PX(JL,4)=PX(JL,4)+YF*FR3-ZF*FR2
      PX(JL,5)=PX(JL,5)+ZF*FR1-XF*FR3
      PX(JL,6)=PX(JL,6)+XF*FR2-YF*FR1
  309 CONTINUE
      WRITE(21)(E(JL),JL=1,NPAN)
  308 CONTINUE
      DO 2424 K=1,6
 2424 WRITE(24)(PX(JL,K),JL=1,NPAN)
      CLOSE(UNIT=21)
      CLOSE(UNIT=24)
C  INVERT E MATRIX
      CALL MATIN(NPAN)
      RETURN
      END
```

```
      SUBROUTINE POTB(PP)
C FIND FORCES AND MOMENTS INDUCED BY TIME RATE OF CHANGE OF SRCE STRNGTH
C    IN SPACE FIXED COORDINATES

      COMMON/BD/XPAN(120),YPAN(120),ZPAN(120),AREA(120),ST(120),
     .ACN(120),ACNV(120),AN(120,3),E(120),P(120,6),PRFS(120),STOLD(120)
     .,PX(120,6)


      COMMON/A/NPAN,NPT,GEE,RHO,NKX,NKY,EYE,DT,TIM,UFWD

      COMPLEX EYE
      DIMENSION PP(6)
      PP(1)=0.00
      PP(2)=0.00
      PP(3)=0.0
      PP(4)=0.00
      PP(5)=0.0
      PP(6)=0.0
      DO 1500 J=1,NPAN
C ST(J) IS TIME RATE OF CHANGE IN HULL-FIXED SYSTEM OF SOURCE STRENGTH
C    OF PANEL J
C  STAV IS AVERAGE SOURCE STRENGTH OVER TIME STEP AT CENTER OF PANEL J
      STAV=STOLD(J)+0.5*DT*ST(J)
C  FIRST TIME STEP IS PURE IMPULSE
      IF(JTM.EQ.1) STAV=0.0
C  TIME DERIVATIVE IN SPACE FIXED SYSTEM
      DO 1200 K=1,6
      PP(K)=PP(K)+(ST(J)*P(J,K)-STAV*UFWD*PX(J,K))*RHO
 1200 CONTINUE
 1500 CONTINUE
      TYPE 1870
 1870 FORMAT('  BODY INDUCE FORCES = ')
      TYPE 2020, PP(1),PP(2),PP(3),PP(4),PP(5),PP(6)
 2020 FORMAT(3X,3F15.6)
      RETURN
      END
```

```
      SUBROUTINE POTST

      COMMON/BD/XPAN(120),YPAN(120),ZPAN(120),AREA(120),ST(120),
     .ACN(120),ACNW(120),AN(120,3),E(120),P(120,6),PRFS(120),STOLD(120)
     .,PX(120,6)

      COMMON/BD2/XPT(150),YPT(150),ZPT(150),WRF(150),WRFR(150)
     .,KK(150,4)

      COMMON/A/NPAN,NPT,GEE,RHO,NKX,NKY,EYE,DT,TIM,UFWD

      COMPLEX A,B,EYE
      DIMENSION XPSL(3,4),XPSLR(3,4),PBB(120,120)
      COMMON/PTST/ARE4(200,4),X4(200,4),Y4(200,4),Z4(200,4)
     .,SEL(200,4)

      DO 1500 J=1,NPAN
      ARE4(J,4)=-1.0
      JT=4
      IF(KK(J,4).EQ.0) JT=3
      DO 1500 JJ=1,JT
      J2=1
      IF(JJ.LT.JT) J2=JJ+1
      KF=KK(J,JJ)
      KG=KK(J,J2)
      X4(J,JJ)=(XPT(KF)+XPT(KG)+XPAN(J))/3.0
      Y4(J,JJ)=(YPT(KF)+YPT(KG)+YPAN(J))/3.0
      Z4(J,JJ)=(ZPT(KF)+ZPT(KG)+ZPAN(J))/3.0
      AF=XPT(KF)-XPAN(J)
      BF=YPT(KF)-YPAN(J)
      CF=ZPT(KF)-ZPAN(J)
      AG=XPT(KG)-XPAN(J)
      BG=YPT(KG)-YPAN(J)
      CG=ZPT(KG)-ZPAN(J)
      CALL SELF(AF,BF,CF,AG,BG,CG,FEE)
      SEL(J,JJ)=FEE
      CR=AF*BG-BF*AG
      AR=BF*CG-CF*BG
      BR=CF*AG-AF*CG
      ARE4(J,JJ)=0.5*SQRT(AR*AR+BR*BR+CR*CR)
 1500 CONTINUE
      DO 127 NJ=1,NPAN
      DO 1277 MJ=1,NPAN
 1277 PBB(NJ,MJ)=0.00
      P(NJ,1)=0.00
      P(NJ,2)=0.00
      P(NJ,3)=0.00
      P(NJ,4)=0.00
      P(NJ,5)=0.00
      P(NJ,6)=0.00
      DO 128 NK=1,4
      ARN=ARE4(NJ,NK)
      IF(ARN.LT.0.0) GO TO 128
      P1=0.00
      P2=0.00
      P3=0.00
      P4=0.00
      P5=0.00
      P6=0.00
      X=X4(NJ,NK)
      Y=Y4(NJ,NK)
      Z=Z4(NJ,NK)
      DO 138 MJ=1,NPAN
      DO 138 MK=1,4
      XF=X4(MJ,MK)
```

```
       YF=Y4(MJ,MK)
      ZF=Z4(MJ,MK)
      ARM=ARE4(MJ,MK)
      IF(ARM.LT.0.00) GO TO 138
      IF(NJ.NE.MJ) GO TO 140
      IF(MK.NE.NK) GO TO 140
      FRA=SEL(MJ,MK)/ARM
      GO TO 1380
  140 RA=SQRT((X-XF)**2+(Y-YF)**2+(Z-ZF)**2)
      FRA=ARM/RA
 1380 RB=SQRT((X-XF)**2+(Y-YF)**2+(Z+ZF)**2)
      FRA=FRA-ARM/RB
      FRX=-AN(NJ,1)*FRA
      FRY=-AN(NJ,2)*FRA
      FRZ=-AN(NJ,3)*FRA
      P1=P1+FRX
      P2=P2+FRY
      P3=P3+FRZ
      P4=P4+YF*FRZ-ZF*FRY
      P5=P5+ZF*FRX-XF*FRZ
      P6=P6+XF*FRY-YF*FRX
      PBB(NJ,MJ)=PBB(NJ,MJ)+FRA*ARN
  138 CONTINUE
      P(NJ,1)=P(NJ,1)+P1*ARN
      P(NJ,2)=P(NJ,2)+P2*ARN
      P(NJ,3)=P(NJ,3)+P3*ARN
      P(NJ,4)=P(NJ,4)+P4*ARN
      P(NJ,5)=P(NJ,5)+P5*ARN
      P(NJ,6)=P(NJ,6)+P6*ARN
  128 CONTINUE
  127 CONTINUE
      OPEN(UNIT=1,FILE='PTSV',ACCESS='SEQOUT',DEVICE='DSK:')
       DO 554 KQ=1,6
  554 WRITE(1)(P(NJ,KQ),NJ=1,NPAN)
       CLOSE(UNIT=1)
      RETURN
      END
```

```
      SUBROUTINE SELF(AF,BF,CF,AG,BG,CG,FEE)
      ASQ=AF*AF+BF*BF+CF*CF
      BSQ=AG*AG+BG*BG+CG*CG
      ADB=AF*AG+BF*BG+CF*CG
      ADB2=ADB+ADB
      ASAS=(AF*BG-BF*AG)**2+(CF*BG-BF*CG)**2+(AF*CG-BF*AG)**2
       FF=0.00
      DO 15 MK=1,10
      DO 15 NK=1,MK
      LA21=NK-MK
      A2SQ=ASQ*LA21*LA21
      DO 15 ML=1,11-MK
      DO 15 NL=1,11-ML
      LB21=NL-ML
       IF(LA21.NE.0) GO TO 5
         IF(LB21.LT.0) GO TO 5
      GO TO 15
    5 R=SQRT(A2SQ+ADB2*LA21*LB21+BSQ*LB21*LB21)
      FF=FF+1.0/R
   15 CONTINUE
      FEE=FF*ASAS*0.002
      RETURN
      END
```

```
      SUBROUTINE MATIN(NPAN)
C INVERTS MATRIX
      DIMENSION E(120,120),BB(120),EST(120)
      OPEN(UNIT=21,FILE='E',ACCESS='SEQIN',DEVICE='DSK:')
      DO 120 J=1,NPAN
  120 READ(21)(E(J,I),I=1,NPAN)
      CLOSE(UNIT=21)
      OPEN(UNIT=21,FILE='E',ACCESS='SEQOUT',DEVICE='DSK:')
      DO 130 J=1,NPAN
      DO 11 MM=1,NPAN
      EST(MM)=0.00
   11 BB(MM)=0.00
      BB(J)=1.0
      EST(J)=1.0/E(J,J)
      DO 17 NIT=1,6
      DO 17 K=1,NPAN
      B=BB(K)
      DO 15 I=1,NPAN
   15 IF(I.NE.K) B=B-E(K,I)*EST(I)
      EST(K)=B/E(K,K)
   17 CONTINUE
      WRITE(21)(EST(K),K=1,NPAN)
  130 CONTINUE
      CLOSE(UNIT=21)
      TYPE 5
    5 FORMAT(' MATINV DONE')
      RETURN
      END
      SUBROUTINE GE(XF,YF,ZF,J,V1,V2,V3,V1R,V2R,V3R,NBT)
      COMMON/BD/XPAN(120),YPAN(120),ZPAN(120),AREA(120),ST(120),
     .ACH(120),ACHW(120),AN(120,3),E(120),P(120,6),PRFS(120),STOLD(120)
     .,PX(120,6)

      COMMON/BD2/XPT(150),YPT(150),ZPT(150),WRF(150),WRFR(150)
     .,KK(150,4)

      COMMON/ARE/RR(500),XZJ(200),YXJ(200),ZYJ(200)

       DIMENSION XSA(3,4),XFA(3),XSAR(3,4)

      J4=J*4
      V1=0.00
      V2=0.00
      V3=0.00
      V1R=0.0
      V2R=0.0
      V3R=0.00
      XNJ=AN(J,1)
      YNJ=AN(J,2)
      ZNJ=AN(J,3)
      NSIDE=4
      IF(KK(J,4).EQ.0) NSIDE=3
      DO 20 JJ=1,NSIDE
      J2=1
      IF(JJ.LT.NSIDE) J2=JJ+1
      J4=J4+1
      KF=KK(J,JJ)
      AF=XPT(KF)
      BF=YPT(KF)
      CF=ZPT(KF)
       R=RR(J4)
      KG=KK(J,J2)
      ANX=(AF-XPT(KG))/R
      ANY=(BF-YPT(KG))/R
      ANZ=(CF-ZPT(KG))/R
      A=AF-XF
```

```
      B=BF-YF
      C=CF-ZF
      TX=XZJ(J)*ANZ-YXJ(J)*ANY
      TY=YXJ(J)*ANX-ZYJ(J)*ANZ
      TZ=ZYJ(J)*ANY-XZJ(J)*ANX
      EX1=A*ANX+B*ANY+C*ANZ
      CALL GO(EX1,R,FF,WRF(KF),WRF(KG))
      V1=V1+FF*TX
      V2=V2+FF*TY
      V3=V3+FF*TZ
       XSA(1,JJ)=-A/WRF(KF)
       XSA(2,JJ)=-B/WRF(KF)
       XSA(3,JJ)=-C/WRF(KF)
       EX1R=EX1+2.0*ZF*ANZ
      CR=-CF-ZF
      CALL GO(EX1R,R,FR,WRFR(KF),WRFR(KG))
      V1R=V1R-FR*TX
      V2R=V2R-FR*TY
      V3R=V3R+FR*TZ
      XSAR(1,JJ)=-A/WRFR(KF)
      XSAR(2,JJ)=-B/WRFR(KF)
      XSAR(3,JJ)=CR/WRFR(KF)
   20 CONTINUE
      G=6.283125307
      IF(J.EQ.NBT) GO TO 84
      CALL SOLID (XSA,G,NSIDE)
      AGG=A*XNJ+B*YNJ+C*ZNJ
      G=-SIGN(G,AGG)
   84 CONTINUE
      CALL SOLID (XSAR,GR,NSIDE)
      AGGR=A*XNJ+B*YNJ-CR*ZNJ
       GR=SIGN(GR,AGGR)
   85 CONTINUE
 7371 FORMAT(' G,GR=',2F15.5)
      V1=V1+XNJ*G
      V2=V2+YNJ*G
      V3=V3+ZNJ*G
      V1R=V1R+XNJ*GR
      V2R=V2R+YNJ*GR
      V3R=V3R-ZNJ*GR
 5590 FORMAT(' V1,V2,V3=',3F15.5)
 5591 FORMAT(' V1R,V2R,V3R=',3F15.5)
      RETURN
      END
```

```
      SUBROUTINE GO(EX1,R,F,RH1,RH2)
      EX2=EX1-R
      SIGN=1.
      IF(R/2..GT.EX1) SIGN=-1.
      UP=SIGN*EX1+RH1
      DN=SIGN*EX2+RH2
      ARG=UP/DN
      X=ARG-1.00
      IF(ABS(X).GT.0.13) GO TO 10
      X2=X*X
      F=SIGN*(X-X2*(.5-.333333333*X+.25*X2))
      RETURN
10    F=SIGN*ALOG(ARG)
      RETURN
      END
```

```
      SUBROUTINE SOLID(XPN,G,NSIDE)
      DIMENSION CS(4),SN(4),Z(4),XPN(3,4)
      G=-6.283185308
      ACR12=XPN(1,1)*XPN(1,2)+XPN(2,1)*XPN(2,2)+XPN(3,1)*XPN(3,2)
      ACR13=XPN(1,1)*XPN(1,3)+XPN(2,1)*XPN(2,3)+XPN(3,1)*XPN(3,3)
      ACR23=XPN(1,2)*XPN(1,3)+XPN(2,2)*XPN(2,3)+XPN(3,2)*XPN(3,3)
      IF(NSIDE.EQ.4) GO TO 40
      G=-3.141592659
      CS(1)=ACR23-ACR13*ACR12
      CS(2)=ACR13-ACR12*ACR23
      CS(3)=ACR12-ACR23*ACR13
      SN(1)=XPN(1,1)*(XPN(2,2)*XPN(3,3)-XPN(3,2)*XPN(2,3))+
     +       XPN(2,1)*(XPN(3,2)*XPN(1,3)-XPN(1,2)*XPN(3,3))+
     +       XPN(3,1)*(XPN(1,2)*XPN(2,3)-XPN(2,2)*XPN(1,3))
      SN(2)=SN(1)
      SN(3)=SN(1)
      SN(4)=0.
      GO TO 50
 40   ACR14=XPN(1,1)*XPN(1,4)+XPN(2,1)*XPN(2,4)+XPN(3,1)*XPN(3,4)
      ACR24=XPN(1,2)*XPN(1,4)+XPN(2,2)*XPN(2,4)+XPN(3,2)*XPN(3,4)
      ACR34=XPN(1,3)*XPN(1,4)+XPN(2,3)*XPN(2,4)+XPN(3,3)*XPN(3,4)
      CS(1)=ACR24-ACR14*ACR12
      CS(2)=ACR13-ACR23*ACR12
      CS(3)=ACR24-ACR34*ACR23
      CS(4)=ACR13-ACR34*ACR14
      B241=XPN(2,2)*XPN(3,4)-XPN(3,2)*XPN(2,4)
      B242=XPN(3,2)*XPN(1,4)-XPN(1,2)*XPN(3,4)
      B243=XPN(1,2)*XPN(2,4)-XPN(2,2)*XPN(1,4)
      B131=XPN(2,1)*XPN(3,3)-XPN(3,1)*XPN(2,3)
      B132=XPN(3,1)*XPN(1,3)-XPN(1,1)*XPN(3,3)
      B133=XPN(1,1)*XPN(2,3)-XPN(2,1)*XPN(1,3)
      SN(1)=XPN(1,1)*B241+XPN(2,1)*B242+XPN(3,1)*B243
      SN(2)=-(XPN(1,2)*B131+XPN(2,2)*B132+XPN(3,2)*B133)
      SN(3)=-(XPN(1,3)*B241+XPN(2,3)*B242+XPN(3,3)*B243)
      SN(4)=XPN(1,4)*B131+XPN(2,4)*B132+XPN(3,4)*B133
 50   CONTINUE
D     TYPE 8844,SN(1),CS(1)
D     TYPE 8844,SN(2),CS(2)
D     TYPE 8844,SN(3),CS(3)
D     TYPE 8844,SN(4),CS(4)
 8844 FORMAT('     SN,CS=',2F15.8)
      SUM=SN(1)+SN(2)+SN(3)+SN(4)
      IF(ABS(SUM).GT.0.01) GO TO 25
      IF(ABS(CS(1)).GT.ABS(SN(1))) GO TO 25
      IF(ABS(CS(2)).GT.ABS(SN(2))) GO TO 25
      IF(ABS(CS(3)).GT.ABS(SN(3))) GO TO 25
 1090 G=SUM*.25
      IF(NSIDE.EQ.3) G=SUM*.166666666667
      RETURN
 25   ST=SN(NSIDE)
      DO 30 I=1,NSIDE
      IF((ABS(CS(I)).LT.9E-8).AND.(ABS(SN(I)).LT..9E-05))
     + GO TO 1090
      IF(ST*SN(I).LT.0.) GO TO 1090
      ST=SN(I)
      C2=CS(I)/SQRT(SN(I)**2+CS(I)**2)
      G=G+ACOS(C2)
 30   CONTINUE
      RETURN
      END
      SUBROUTINE PREP(J)


      COMMON/BD/XPAN(120),YPAN(120),ZPAN(120),AREA(120),ST(120),
     .ACN(120),ACNW(120),AN(120,3),E(120),P(120,6),PRFS(120),STOLD(120)
     .,PX(120,6)
```

```
      COMMON/BD2/XPT(150),YPT(150),ZPT(150),WRF(150),WRFR(150)
     .,KK(150,4)
      COMMON/ARE/RR(500),XZJ(200),YXJ(200),ZYJ(200)
      ZYT=0.0
      YXT=0.00
      XZT=0.00
      J4=J*4
      JT=4
      IF(KK(J,4).EQ.0) JT=3
      DO 20 JJ=1,JT
      J4=J4+1
       J2=1
      IF(JJ.LT.JT) J2=JJ+1
       KF=KK(J,JJ)
      KG=KK(J,J2)
      AG=XPT(KG)
      BG=YPT(KG)
      CG=ZPT(KG)
      AF=XPT(KF)
      BF=YPT(KF)
      CF=ZPT(KF)
      R=SQRT((AF-AG)**2+(BF-BG)**2+(CF-CG)**2)
      XT=AF-XPAN(J)
      YT=BF-YPAN(J)
      ZT=CF-ZPAN(J)
      ANX=(AF-AG)/R
      ANY=(BF-BG)/R
      ANZ=(CF-CG)/R
      DOT=ANX*XT+ANY*YT+ANZ*ZT
      XT=XT-DOT*ANX
      YT=YT-DOT*ANY
      ZT=ZT-DOT*ANZ
      ZYT=ZYT+ZT*ANY-ANZ*YT
      YXT=YXT+YT*ANX-ANY*XT
      XZT=XZT+XT*ANZ-ANX*ZT
      RR(J4)=R
   20 CONTINUE
      XZJ(J)=SIGN(AN(J,2),XZT)
      YXJ(J)=SIGN(AN(J,3),YXT)
      ZYJ(J)=SIGN(AN(J,1),ZYT)
      RETURN
       END
```